

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-198231  
 (43)Date of publication of application : 31.07.1997

(51)Int.Cl. G06F 7/00  
 G06F 9/305  
 G06F 15/16

(21)Application number : 08-027264 (71)Applicant : NEC CORP  
 (22)Date of filing : 22.01.1996 (72)Inventor : INOUE TOSHIAKI

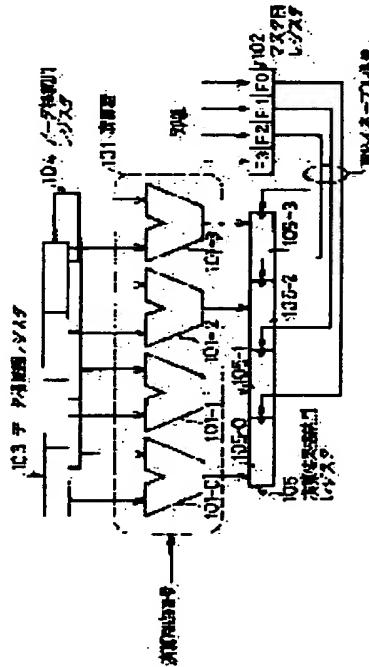
## (54) ARITHMETIC PROCESSOR

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To perform a mask processing at high speed in the arithmetic processor provided with plural arithmetic units performing the same arithmetic processing in parallel.

**SOLUTION:** In each of bits F0 to F3 of a register 102 for mask, the mask information showing whether the fields 105-0 to 105-3 of a register 105 for storage of arithmetic result are masked or not is stored. The register 102 for mask outputs the write enable signal in accordance with the mask information stored in each of bits F0 to F3 to each of fields 105-0 to 105-3.

Therefore, the mask processing controlling whether the arithmetic results of arithmetic units 101-0 to 101-3 are written in the fields 105-0 to 105-3 or not can be performed having only the mask information written in the register 102 for mask.



## LEGAL STATUS

[Date of request for examination] 22.01.1996

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2806346

[Date of registration] 24.07.1998

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-198231

(43)公開日 平成9年(1997)7月31日

(51) Int.Cl. <sup>6</sup>	識別記号	府内整理番号	F I	技術表示箇所
G 06 F 7/00			G 06 F 7/00	H
9/305			15/16	3 9 0 Z
15/16	3 9 0		9/30	3 4 0 A

審査請求 有 請求項の数 3 FD (全 8 頁)

(21)出願番号 特願平8-27264

(22)出願日 平成8年(1996)1月22日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 井上 傑明

東京都港区芝五丁目7番1号 日本電気株式会社内

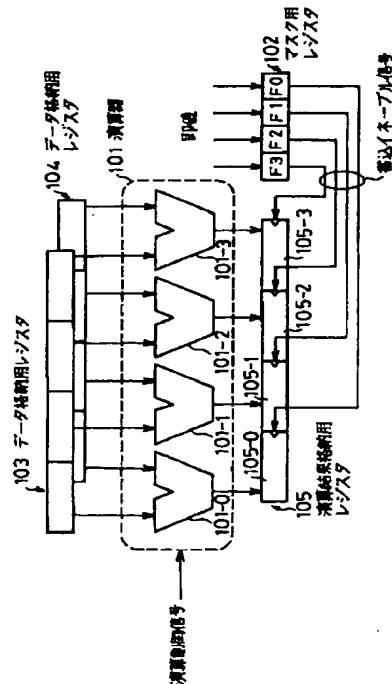
(74)代理人 弁理士 境 廣巳

(54)【発明の名称】 演算処理装置

(57)【要約】

【課題】 同一の演算処理を並行して行う複数個の演算器を備えた演算処理装置に於いて、マスク処理を高速に行えるようにする。

【解決手段】 マスク用レジスタ102の各ビットF0～F3には、演算結果格納用レジスタ105のフィールド105-0～105-3をマスクするか否かを示すマスク情報が格納されており、マスク用レジスタ102は、各ビットF0～F3に格納されているマスク情報に従った書き込みイネーブル信号を各フィールド105-0～105-3に対して出力する。従って、マスク用レジスタ102にマスク情報を書き込んでおくだけで、演算器101-0～101-3の演算結果をフィールド105-0～105-3に書き込むか否かを制御するといったマスク処理を行うことができる。



## 【特許請求の範囲】

【請求項1】 同一の演算処理を並行して行う複数個の演算器を備えた演算処理装置に於いて、書き込みイネーブル信号によって書き込み制御が可能な、前記各演算器毎の演算結果格納用レジスタと、該各演算結果格納用レジスタ毎のマスク情報を保持し、保持している各マスク情報に従って、前記各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みイネーブル信号を出力するマスク用レジスタとを備えたことを特徴とする演算処理装置。

【請求項2】 同一の演算処理を並行して行う複数個の演算器を備えた演算処理装置に於いて、書き込みイネーブル信号によって書き込み制御が可能な、前記各演算器毎の演算結果格納用レジスタと、前記各演算器が生成する演算フラグに基づいて、前記各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みイネーブル信号を出力する論理回路とを備えたことを特徴とする演算処理装置。

【請求項3】 前記演算器が生成する演算フラグは、演算結果の符号を示すものであることを特徴とする請求項2記載の演算処理装置。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、マイクロプロセッサ等に使用される演算処理装置に関し、特に、単一の命令によって制御される同一構造の演算器を複数備えた演算処理装置に関する。

## 【0002】

【従来の技術】 従来、RISC (Reduced Instruction Set Computer) 型の汎用マイクロプロセッサでは、1命令で処理できるワード長を増やしたり（高精度化）、RISC命令の実行パイプラインの段数を増加させて動作周波数を向上させたり（スーパー・パイプライン化）、或いは演算器等を複数個搭載して演算処理を並列実行する（スーパースカラ化）等の工夫をすることによって、整数演算や浮動小数点演算の処理性能の向上を図ってきた。一方、JPEG (Joint Photographic Experts Group) やMPEG (Moving Picture Experts Group) 等の画像圧縮伸張処理や、3Dレンダリング処理に代表される画像処理では、画素データの精度（ダイナミックレンジ）は高々8ビットから16ビット程度である。従って、これらの画像処理を、従来のRISCプロセッサであるR4000や21164等で実行する場合、データの必要精度に関係なく、64ビットの演算器を用いて1画素ずつ処理を行わなければならず、非効率的であった。

【0003】 これに対して、UltraSPARCやPA 50

7100LC等のRISCプロセッサでは、画素データを処理する専用の演算器や命令セットを拡張することによって、画像処理を効率良く実行できるようしている。処理単位が64ビットのUltraSPARCや処理単位が32ビットのPA7100LCでは、64ビットまたは32ビットの演算器を、8ビットまたは16ビットという低精度の複数個の演算器に分割し、更に、各演算器の演算結果を格納するレジスタに関しても、それを各演算器の精度に応じたビット幅を有する複数のフィールドに分割することにより、複数の画像データを並列に演算処理できるようにしたSIMD (Single Instruction Multiple Data) 構造を採用している。このようにすることにより、R4000や21164と比較して画像データに対する演算処理性能は向上される。

【0004】 図4は従来のSIMD構造を採用したプロセッサの演算処理装置の構成例を示したブロック図であり、演算器101と、演算対象となるデータが格納されるデータ格納用レジスタ103, 104と、演算結果格納用レジスタ401とから構成されている。

【0005】 演算器101は、外部から与えられる演算制御信号によってより低い精度の4個の演算器101-0～101-3として動作可能な構造を有する。データ格納用レジスタ103, 104及び演算結果格納用レジスタ401は、それぞれ演算器101-0～101-3に対応する4個のフィールドに分割されている。このように、演算器101を4個の演算器101-0～101-3に分割すると共に、データ格納用レジスタ103, 104をそれぞれ4個のフィールドに分割することにより、各フィールドに格納されているデータをそれぞれ独立したデータと見做して4個の演算器101-0～101-3で並列に同一の演算処理を行うことができる。尚、各演算器101-0～101-3の演算結果は、演算結果格納用レジスタ401の対応するフィールドに格納される。

## 【0006】

【発明が解決しようとする課題】 上述した従来のSIMD構造を採用したプロセッサの演算処理装置は、データ格納用レジスタの各フィールドに格納されたデータに対して同一の演算処理を行い、演算結果を演算結果格納用レジスタの各フィールドに直接書き込む構造であるため、演算結果格納用レジスタの特定のフィールドを新たな演算結果で更新し、他のフィールドは古い値のままにするといったマスク処理を行う場合、多くの命令が必要となる。画像処理の場合、奥行き（Z値）判定処理、画像の奥行きに応じた重ね合わせ処理、動き補償予測処理等に於いて、マスク処理は多用されるため、マスク処理のために多くの命令が必要になる従来の演算処理装置では処理速度の大幅な低下を招いてしまう。

【0007】図5は、図4に示した演算処理装置にマスク処理を行わせるために必要となる命令列を示した図である。

【0008】図5に於いて、第1番目の命令「ADD4 r4, r0, r3」は、レジスタr3(図4のレジスタ101に対応)に格納されている古い値と常に0を保持しているレジスタr0の内容とを加算し、レジスタr4に格納することを指示する命令である。

【0009】第2番目の命令「ADD4 r3, r1, r2」は、レジスタr1, r2(図4のレジスタ103, 104に対応)に格納されている値を加算し、加算結果をレジスタr3に格納することを指示する命令である。

【0010】第3番目の命令「ADD4 r5, r0, imm」は、即値immとレジスタr0の内容とを加算することによりマスクパターンを生成し、生成したマスクパターンをレジスタr5に格納することを指示する命令である。

【0011】第4番目の命令「AND r3, r3, r5」は、レジスタr3, r5の論理積をとり、その結果をレジスタr3に格納することを指示する命令である。

【0012】第5番目の命令「NOR r5, r0, r5」は、レジスタr0, r5の否定論理和をとり、その結果をレジスタr5に格納することを指示する命令である。

【0013】第6番目の命令「AND r4, r4, r5」は、レジスタr4, r5の論理積をとり、その結果をレジスタr4に格納することを指示する命令である。

【0014】第7番目の命令「OR r3, r3, r4」は、レジスタr3, r4の論理和をとり、その結果をレジスタr3に格納することを指示する命令である。

【0015】以上の7個の命令を実行することにより、マスク処理が行われ、その結果がレジスタr3に格納される。

【0016】尚、SIMD構造を採用したプロセッサの演算処理装置としては、特願平5-189585号公報に記載されている演算処理装置のように、演算制御信号入力を持たず、複数の演算器を最初から備えたものも知られているが、特願平5-189585号公報に記載されている演算処理装置も、図4に示した演算処理装置と同様に、各演算器の演算結果をレジスタに直接書き込む構造であるので、上述したと同様の問題点がある。

【0017】そこで、本発明の目的は、マスク処理を行うために必要となる命令数を少なくすることができる演算処理装置を提供することにある。

【0018】

【課題を解決するための手段】本発明は上記目的を達成するため、同一の演算処理を並行して行う複数個の演算器を備えた演算処理装置に於いて、書き込み信号によって書き込み制御が可能な、前記各演算器毎の演算

結果格納用レジスタと、該各演算結果格納用レジスタ毎のマスク情報を保持し、保持している各マスク情報に従って、前記各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みインターブル信号を出力するマスク用レジスタとを備えたものである。

【0019】この構成に於いては、マスク用レジスタが、保持している各演算結果格納用レジスタ毎のマスク情報に従って、各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みインターブル信号を出力する。

【0020】また、本発明は、各演算器の演算結果に従ってマスク処理を行うことを可能にすると共に、その際の命令数を少なくするため、同一の演算処理を並行して行う複数個の演算器を備えた演算処理装置に於いて、書き込みインターブル信号によって書き込み制御が可能な、前記各演算器毎の演算結果格納用レジスタと、前記各演算器が生成する演算フラグに基づいて、前記各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みインターブル信号を出力する論理回路とを備えたものである。

【0021】この構成に於いては、論理回路が、各演算器で生成された演算フラグに基づいて、各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みインターブル信号を出力する。

【0022】

【発明の実施の形態】次に本発明の実施の形態について図面を参照して詳細に説明する。

【0023】図1は本発明の一実施例のブロック図であり、演算器101と、マスク用レジスタ102と、データ格納用レジスタ103, 104と、演算結果格納用レジスタ105とから構成されている。

【0024】演算器101は、外部から与えられる演算制御信号によって4個の演算器101-0～101-3として並列動作可能な構造を有している。

【0025】データ格納用レジスタ103, 104は、それぞれ演算器101-0～101-3に対応する4個のフィールドに分割されており、各フィールドには演算器101-0～101-3で演算を行うデータが格納される。

【0026】演算結果格納用レジスタ105は、演算器101-0～101-3に対応する4個のフィールド105-0～105-3に分割されている。また、各フィールド105-0～105-3は書き込みインターブル信号によって個別に書き込み制御が可能になっている。

【0027】マスク用レジスタ102は、4ビット構成のレジスタであり、各ビットF0～F3には、演算結果格納用レジスタ105の各フィールド105-0～105-3をマスクするか否かを示すマスク情報が格納される。また、マスク用レジスタ102は、各ビットF0～

F 3 の値に従った書込イネーブル信号を、各フィールド 105-0~105-3 に対して出力する。

【0028】このような構成をとることにより、図5の命令列によって行われるマスク処理と同等のマスク処理を、図2に示す命令列で実行することが可能になる。

【0029】図2に於いて、第1番目の命令「SET r4, imm」は、レジスタ r4 (図1のマスク用レジスタ 102 に対応) にマスク処理のためのビット列 imm を格納することを指示する命令であり、第2番目の命令「ADD4 r3, r1, r2, r4」は、レジスタ r4 の内容に従って演算結果格納用レジスタ 105 の各フィールド 105-0~105-3 をマスクすることを指示すると共に、レジスタ r1, r2 (図1のデータ格納用レジスタ 103, 104 に対応) の内容を加算して加算結果をレジスタ r3 (図1の演算結果格納用レジスタ 105 に対応) に格納することを指示する命令である。このように、本実施例によれば、図5に示した7個の命令から構成される命令列を、図2に示すような2個の命令から構成される命令列で置き換えることが可能になるため、マスク処理を実行するために必要となる命令数を約70%削減できる。

【0030】次に、現画面の奇数ライン上の画素データと前画面の奇数ライン上の画素データとの誤差、及び現画面の偶数ライン上の画素データと前画面の偶数ライン上の画素データとの誤差をそれぞれ求める場合を例にとって、本実施例の動作を説明する。尚、上述したような処理は、MPEG 等の動画像圧縮処理の動き補償予測処理に於いて行われる処理である。

【0031】先ず、SET 命令を用いてマスク用レジスタ 102 の各ビット F0, F1, F2, F3 にそれぞれ “1”, “0”, “1”, “0” のマスクパターンを格納する。これにより、マスク用レジスタ 102 から各フィールド 105-0~105-3 に供給される書込イネーブル信号がそれぞれ “1”, “0”, “1”, “1” となり、フィールド 105-1, 105-3 に対する演算結果の書き込みがマスクされる。

【0032】次いで、演算結果格納用レジスタ 105 の各フィールド 105-0~105-3 に 0 を格納すると共に、図示を省略した累積値保存用レジスタに 0 を格納する。

【0033】その後、データ格納用レジスタ 103 の4個のフィールドに、それぞれ現画面の第1ライン～第4ライン上の第1番目の画素データを書き込むと共に、データ格納用レジスタ 104 の4個のフィールドに、それぞれ前画面の第1ライン～第4ライン上の第1番目の画素データを書き込む。

【0034】次いで、各演算器 101-0~101-3 に減算動作を行わせる。これにより、各演算器 101-0~101-3 から、前画面の第1ライン～第4ラインの第1番目の画素データと現画面の第1ライン～第4ラ

インの第1番目の画素データとの誤差が output される。

【0035】この時、演算結果格納用レジスタ 105 のフィールド 105-1, 105-3 は、書込イネーブル信号によってマスクされているので、演算器 101-1, 101-3 から出力される第2ライン、第4ラインの誤差はフィールド 105-1, 105-3 に格納されず、演算器 101-0, 101-2 から出力される第1ライン、第3ラインの誤差のみがフィールド 105-0, 105-2 に格納される。

10 【0036】その後、図示を省略した演算手段によって演算結果格納用レジスタ 105 の各フィールド 105-0~105-3 に格納されている誤差を累算し、更に、累積値保存用レジスタに保存されている累積値と上記累算結果とを加算した値を累積値保存用レジスタに格納する。

【0037】次いで、データ格納用レジスタ 103 の4個のフィールドに、それぞれ現画面の第1ライン～第4ライン上の第2番目の画素データを書き込むと共に、データ格納用レジスタ 104 の4個のフィールドに、それ

20 それ前画面の第1ライン～第4ライン上の第2番目の画素データを書き込み、以下前述したと同様の処理を全ラインについて行う。この時、レジスタ 105 のフィールド 105-1, 105-3 はマスクされ、0 を保持しているので、上記した処理を行うことにより、現画面の奇数ライン上の画素データと前画面の奇数ライン上の画素データとの誤差を求めることが可能である。

【0038】その後、現画面の偶数ライン上の画像データと前画面の偶数ライン上の画像データとの誤差を求めるために、SET 命令を用いてマスク用レジスタ 102 の各ビット F0, F1, F2, F3 にそれぞれ “0”, “1”, “0”, “1” のマスクパターンを格納する。これにより、マスク用レジスタ 102 から各フィールド 105-0~105-3 に供給される書込イネーブル信号がそれぞれ “0”, “1”, “0”, “1” となり、フィールド 105-0, 105-2 がマスクされる。

【0039】以後、前述したと同様の処理を全てのラインについて行う。この場合、レジスタ 105 のフィールド 105-0, 105-2 はマスクされ、0 を保持しているので、上記した処理を行うことにより、現画面の偶数ライン上の画素データと前画面の偶数ライン上の画素データとの誤差を求めることが可能である。

【0040】次に本発明の他の実施例について説明する。図3は本発明の他の実施例のブロック図であり、演算器 201 と、論理回路 202 と、データ格納用レジスタ 103, 104 と、演算結果格納用レジスタ 105 とから構成されている。

【0041】演算器 201 は、外部から与えられる演算制御信号によって4個の演算器 201-0~201-3 として並列動作可能な構成を有する。また、各演算器 201-1~203-3 はそれぞれ演算フラグを生成する

構造を有している。ここで、演算フラグとは、符号フラグ、ゼロフラグ等である。

【0042】データ格納用レジスタ103、104は、それぞれ演算器201-0～201-3に対応する4個のフィールドに分割されており、各フィールドには対応する演算器201-0～201-3で演算を行うデータが格納される。

【0043】演算結果格納用レジスタ105は、演算器201-0～201-3に対応する4個のフィールド105-0～105-3に分割されている。また、各フィールド105-0～105-3は書き込みイネーブル信号によって個別に書き込み制御が可能になっている。

【0044】論理回路202は、各演算器201-0～201-3が生成する演算フラグに対して論理演算を行い、演算結果格納用レジスタ105の各フィールド105-0～105-3に対して演算器201-0～201-3の演算結果を書き込むか否かを制御する書き込みイネーブル信号を出力する機能を有する。

【0045】次に、3Dレンダリング処理に於ける奥行き(Z値)判定処理を例にとって本実施例の動作を説明する。

【0046】先ず、現在スクリーンに表示されている第1ラインの第1番目～第4番目の画素のZ値を、データ格納用レジスタ103、演算結果格納用レジスタ105の各フィールドに格納すると共に、次に表示しようとする画面の第1ラインの第1番目～第4番目の画素のZ値をデータ格納レジスタ104の各フィールドに格納する。

【0047】その後、各演算器201-0～201-3が、データ格納用レジスタ104の各フィールドに格納されているZ値からデータ格納用レジスタ103の対応するフィールドに格納されているZ値を減算し、減算結果の符号を演算フラグとして出力する。

【0048】論理回路202は、各演算器201-0～201-3から出力された演算フラグに基づいて演算結果格納用レジスタ105の各フィールド105-0～105-3に対する書き込みイネーブル信号を生成する。その際、論理演算回路201-0～201-3から出力される演算フラグが負を示すものである場合には、各フィールド105-0～105-3へ出力する書き込みイネーブル信号を“1”にし、正を示すものである場合には、書き込みイネーブル信号を“0”にする等の論理演算を行う。

【0049】今、例えば、論理回路202から各フィールド105-0～105-3に出力される書き込みイネーブル信号の内、フィールド105-0に対する書き込みイネーブル信号が“1”、他のフィールド105-1～105-3に対する書き込みイネーブル信号が“0”になったとする。

【0050】その後、各演算器201-0～201-3が、データ格納用レジスタ104の各フィールドに格納

されているZ値を出力する。

【0051】この時、各フィールド105-0～105-3に対する書き込みイネーブル信号の内、フィールド105-0に対する書き込みイネーブル信号のみが“1”となっているので、演算器201-0～201-3の演算結果の内、演算器201-0の演算結果のみがレジスタ105のフィールド105-0に書き込まれる。次いで、演算結果格納用レジスタ105の4個のフィールド105-0～105-3に格納されているZ値を、第1ラインの第1番目～第4番目の画素の最新のZ値として所定の記憶領域に保存する。

【0052】その後、現在スクリーンに表示されている第1ラインの第5番目～第8番目の画素のZ値を、データ格納用レジスタ103、演算結果格納用レジスタ105の各フィールドに格納すると共に、次に表示しようとする画面の第1ラインの第5番目～第8番目の画素のZ値をデータ格納レジスタ104の各フィールドに格納し、以後、前述したと同様の処理を行う。以上の処理を全ての画素について繰り返し行う。

【0053】このように、本実施例によれば、演算器201-0～201-3の演算結果に従ってマスクするフィールドを選択する際、条件分岐命令を使用する必要がなくなり、条件分岐命令実行に伴うオーバヘッドを皆無にすることができるので、マスク処理を高速化することができる。

【0054】尚、上述した実施例は、演算器を4つに分割した場合について説明したが、本発明はこれに限られるものではない。また、演算制御信号入力を持たない、複数の演算器を最初から備えた演算処理装置に対しても、本発明を適用できることは勿論である。

【0055】

【発明の効果】以上説明したように、本発明は、各演算結果格納用レジスタ毎のマスク情報を格納し、格納している各マスク情報に従って、各演算結果格納レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する信号を出力するマスク用レジスタを備えているので、マスク処理を行うために必要となる命令数を少なくし、3Dレンダリング処理等のマスク処理を必要とする画像処理を高速化することができる。

【0056】また、本発明は、各演算器で生成された演算フラグに基づいて、各演算結果格納用レジスタに対して対応する演算器の演算結果を書き込むか否かを制御する書き込みイネーブル信号を出力する論理回路を備えているので、各演算器の演算結果に従ったマスク処理を条件分岐命令を実行することなく行え、その結果、処理速度を高速化することができる。

【図面の簡単な説明】

【図1】本発明の一実施例のブロック図である。

【図2】図1の実施例に於いてマスク処理を行う際に必要となる命令列を示した図である。

【図3】本発明の他の実施例のブロック図である。

【図4】従来の技術の一例を示すブロック図である。

【図5】図4に示した従来の技術に於いて、マスク処理を行う際に必要となる命令列を示した図である。

【符号の説明】

101, 101-0~101-3, 201, 201-0\*

\* ~ 201-3 … 演算器

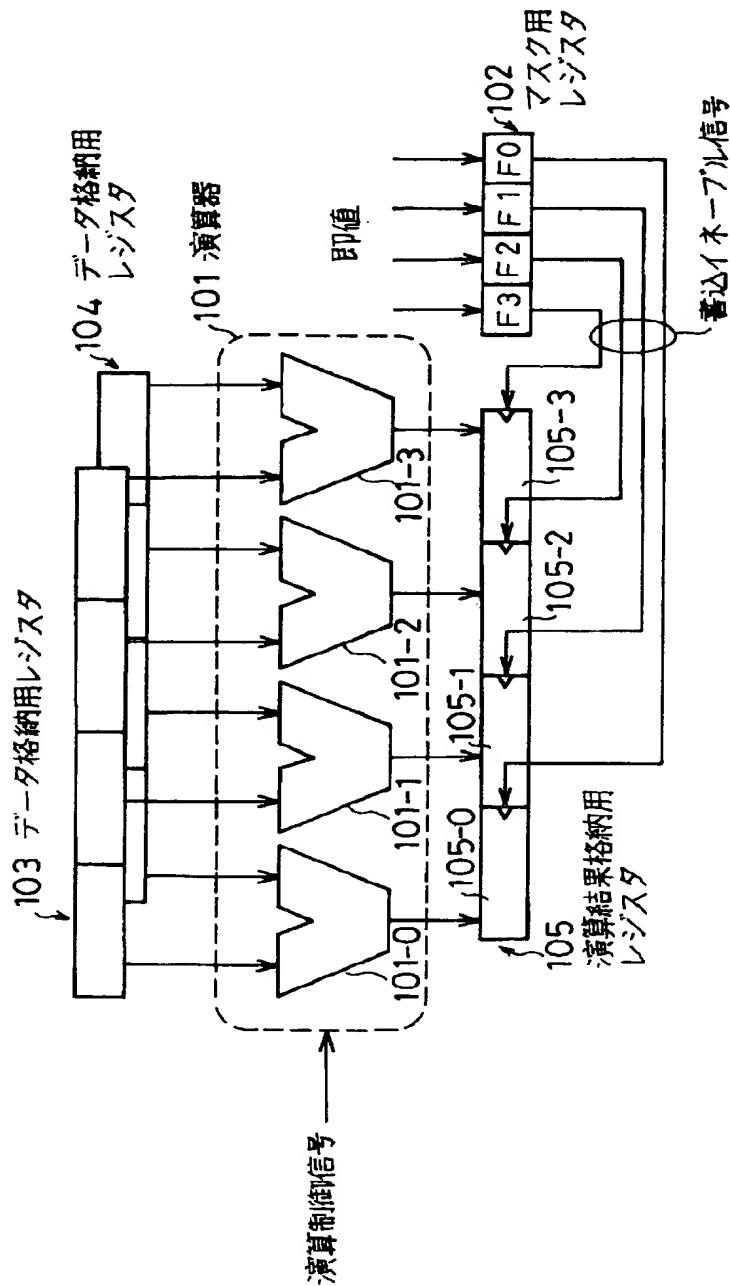
102 … マスク用レジスタ

103, 104 … データ格納用レジスタ

105, 401 … 演算結果格納用レジスタ

202 … 論理回路

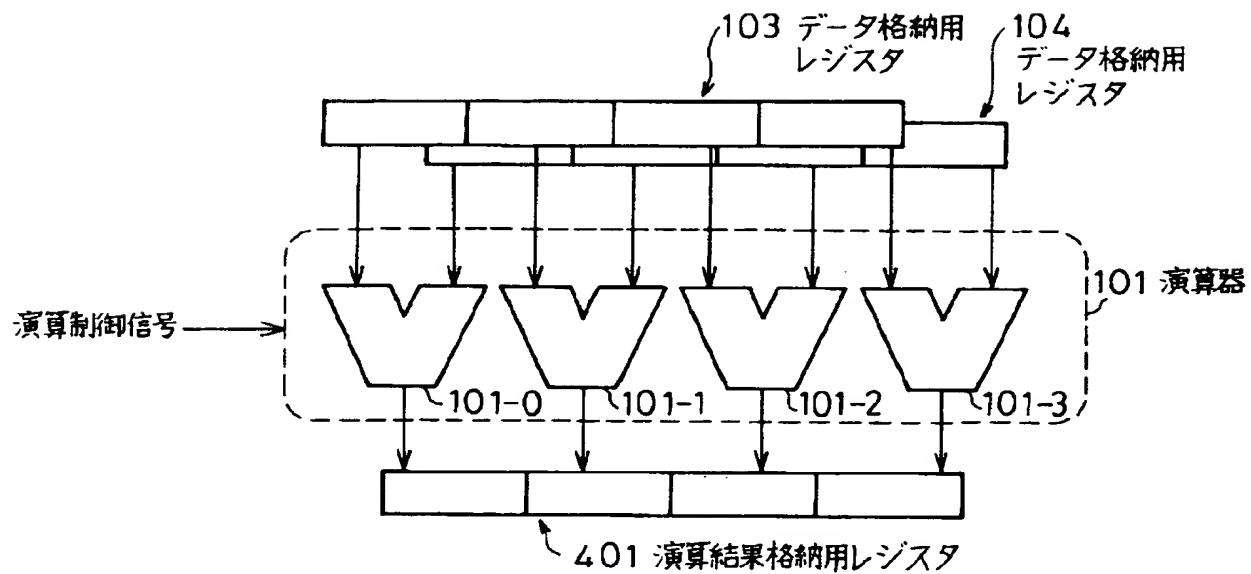
【図1】



【図2】

1: SET r4, imm  
 2: ADD4 r3, r1, r2, r4

【図4】



【図5】

1: ADD4 r4, r0, r3 . . . 古い値のコピー  
 2: ADD4 r3, r1, r2 . . . 通常の加算  
 3: ADD4 r5, r0, imm . . . マスクパターンの生成  
 4: AND r3, r3, r5  
 5: NOR r5, r0, r5  
 6: AND r4, r4, r5  
 7: OR r3, r3, r4

マスク演算結果の生成

【図3】

